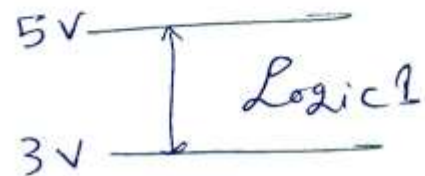


micro sec 5

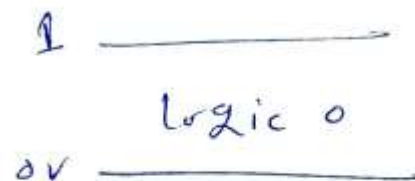
from slide 41 (الجزء ٥٥٥٥)

before this is orientation.

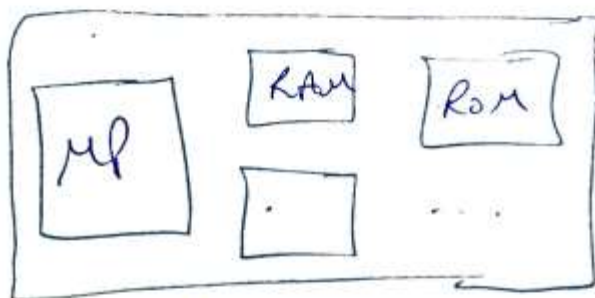


دو ایم معلومہ فی اول ای صفحہ۔

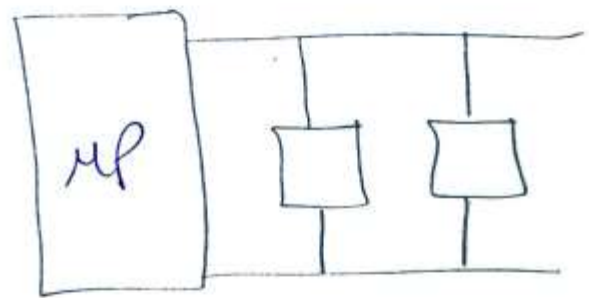
Gap



Difference between MC and MP.



MC



Difference between ES & PC.

(ES) - Embedded System (MC) - Embedded System (task) - Embedded System

(PC) - Personal Computer (MP) - Personal Computer (task) - Personal Computer

Criteria for choosing Mc

- * Speed
- * Power Consumption
- * Packaging → (Mc) في أي (Package)
- * Cost
- * RAM & ROM.
- * I/O Ports.

Community

دی (issue) معاً اختیار ال (micro-controller)
لا تيجر مثلاً تختار (IDE) ويعجل مثلاً.
لما بتقدر على الحاجة المشغولة ودي نقطة قهنا.

Components of 8051 Mc

- a) 128 bytes RAM.
- b) 4 K ROM.
- c) 2 timers.
- d) one serial Port
- e) 4 I/O Ports (each 8 bits)
- f) 6 interrupt.

8051 Assembly Language Programming

• $\text{نوع (instructions)}$ 11 نوع (Registers) 8

(Registers) 8

a) 8-bit registers Mc

b) A: Accumulator, must be destination of any operation.

c) B, PC, DPTR, SP.

d) R0-7.

ex: Mov Destination, source
• نوع (Data) نوع (Data)

~~Move~~ Mov R1, #5H \rightarrow immediate [FFH - 255]
 \hookrightarrow max value of Register.

Mov R1, R2 \rightarrow Register.

Mov R1, #255

\hookrightarrow illegal instruction.

~~Cause 255F is max. value of reg.~~

[ex] ADD A, R1

ADD A, #34F

ADD R2, A → x x

Any Assembly Program is

- instructions

- Directives

~~Comments~~

← مثل أوامر مظهر في الكود.

لكنها تساعد ال (Assembler) على عمل (Assembly)

instructions [label:] opCode [operands] [Comments]

له الشكل العام ليه.

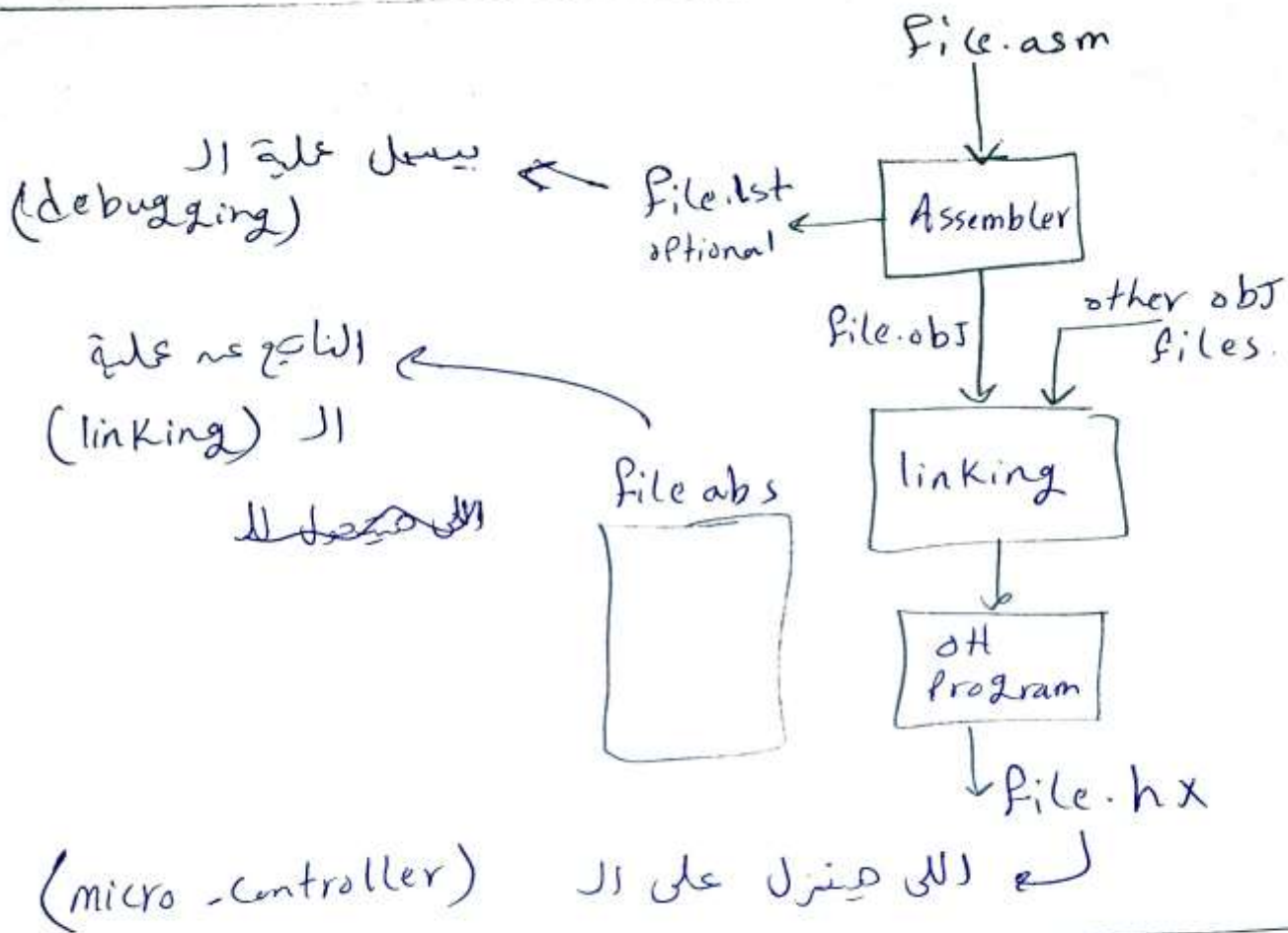
المرحلة

(File.asm) يكتب فيها ال (instruction) التي عايزها.

له بيحصل على ال (assembler) يطلع (File.object)

(File.object) التالي الينا محتاج يحصل ليه (linking)

مع الحاجة الينا عايز تعلم.



shape of File.lst

Address	Machine Code	Assembly	Comments
0000	<u>7D</u> <u>25</u>	ORG 0000 Mov R5, #25	
0002	7F34	Mov R6, #34	

0000	7D
0001	25
0002	7F
0003	34
⋮	⋮

ROM

5

(directive)

Address	Machine.	Assem.
0004	2D	ADD A, R5
0005	-	-

← أول ما ننتج ال (Program) فتلقا له ال
 (Program Counter) عندك فيه $PC = 0000$
 16 bit register.

وكل شوية نعمل (increment) وبتبقى عارف انه اللي جاي
 صحت (instruction).

له مقدرة اكتب برنامج الذاكرة بتاخذ اكثر من (4 K byte)
 لأن سعة ال (RAM) حدى (4 K byte)

Directives

← ليست أوامر

→ هدف مساعدة ال (Assembler).

ex ORG address $\Rightarrow PC = \text{Address}$

له معناها انه البرنامج بتاعى ال (instruction) هيق
 موجود فى ال (Address) اللي عنده ده وهو
 اللي ال (Program Counter) هيشير عليه فى ال اول

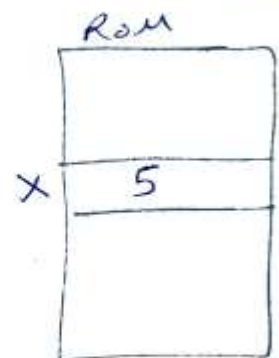
Ex END

له بغير ~~ال~~ ال (Assembler) له دى نفاة الجزء
الى بغير (Assembly).

Ex EQU

له عندى (Constant) و عايزه يتصافى مكانه فى ال (Rom)

Const X = 5;



instead

$$\left. \begin{array}{l} y = x + 3 \\ y = 5 + 3 \end{array} \right\} \text{Compilation (error)}$$

لأنه X مش معرف

له قبل المرحلة دى فيه مرحلة (Pre Processor)

له كتل (`#define x`) عشان فى الكود كل ما يشوف

X يحلها ب 5.

له بعد (declaration) لا (Constant) فيه غير ما

احجز له مكانه فى ال (memory)

Count EQU, 25

Mov R3, ~~#~~Count

Mov R3, Count (x x)

* DB:

له نوعان أصلي (Assembly) بار Hello world
له محتاج تكون موجودة في ال (memory)

(label) يشتر على ال
(string)

ORG 500

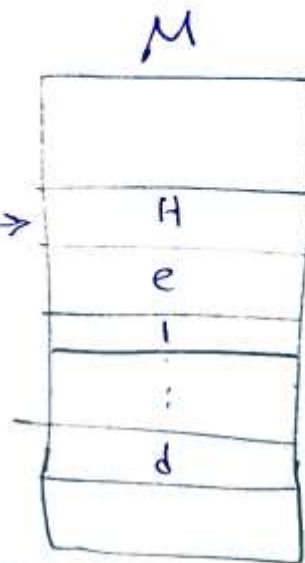
Data1 DB "Hello world"

عند ال (label data1) الى هو عبارة عن 500

سأني يقول انه يشار على الأمر ده.

له أنا عرفت بار (DB) انه مجموعة الأوامر اللي محتجوزة

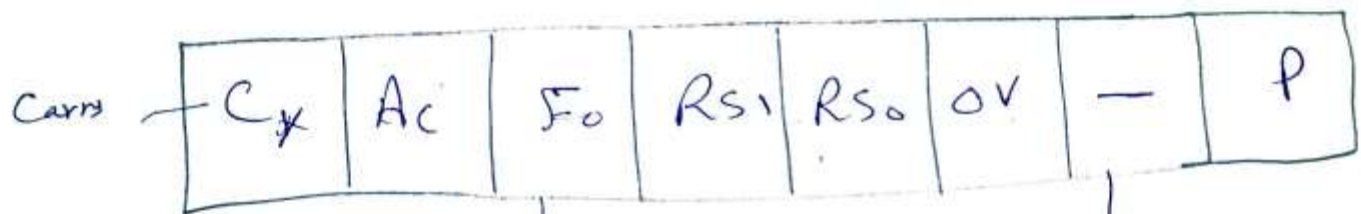
من أول ال (address 500) محتجوزة ل Hello world.



* Flag Register (PSW)

↳ Program status word

6-bit ~~في الذاكرة~~ (8-bit register) ~~مستخدم~~



↓
not used.

P = 1 ~~عند الحساب~~ odd
P = 0 even

user defined

P → Parity (1 or 0) OV → over flow

~~Ac~~ Ac → Auxiliary Carry

~~Cy~~ Cy → Carry

$$\begin{array}{r}
 00111000 \\
 00101111 \\
 \hline
 01100111 \\
 \hline
 \begin{array}{cc}
 \underline{6} & \underline{7}
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 38 \\
 + 2F \\
 \hline
 67
 \end{array}$$

P = 1 , Ac = 1 → ~~النتيجة مرتين~~ ~~في الذاكرة~~

Look at
Page 86

* RAM: 128 bytes

متغیروں اور آپریشنز کے لیے
(variables) اور آپریشنز کے لیے

(RS1, RS0)

Bank selectors

حسب قیسم بکٹس (Bank)
کی (register) بتائی
جائیں گی

00 → Bank 0 01 → Bank 1

10 → Bank 2 11 → Bank 3

(Bank 1) کی default بتائی جائے گی (stack)

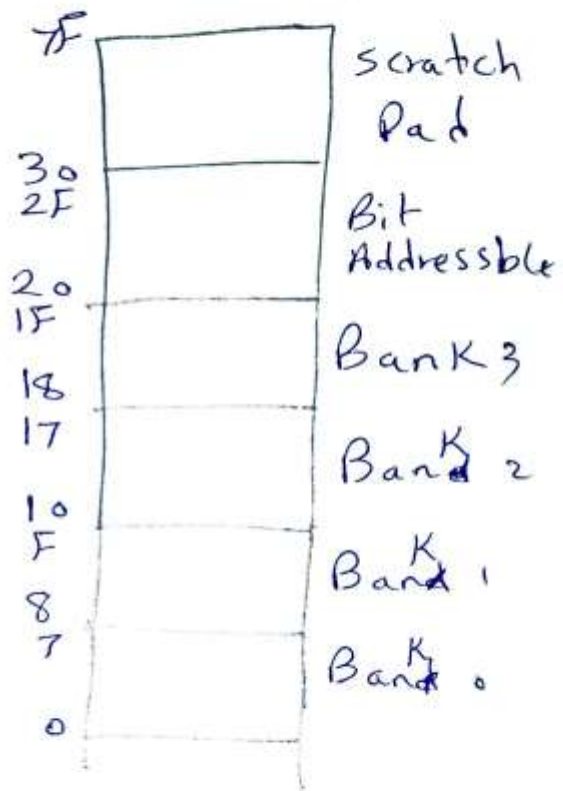
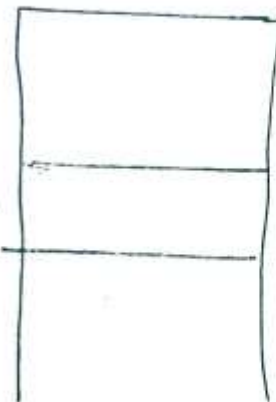
pop, push (stack) کے لیے

Default

SP = 7

پتہ (Bank 1) بتائی
جائے گی (8)

SP
↓
8-bit
Reg



Push \rightarrow increment then push

Pop \rightarrow pop then decrement .

Mov R1, #05H

Push R1

— لا يشترط أن ال (SP=7) هو فقط يساوي

على عنوان معين ويكون العنوان الذي بعده هو الذي

يخزن فيه ال (data) بتاخر .

11